

## TCP SACK 突发分组丢失吞吐量模型

王志明, 曾孝平, 李娟, 刘学, 陈礼

(重庆大学通信工程学院, 重庆 400044)

**摘要:** 利用 Gilbert 分组丢失模型描述端对端突发分组丢失特性, 提出了基于 RFC6675 的快重传和快恢复模型, 推导并基于该模型建立 TCP SACK 吞吐量模型。数值实验和仿真实验表明, 快重传和快恢复模型能准确描述基于 RFC6675 的快重传和快恢复过程; TCP SACK 流吞吐量模型估计的准确性得到提升。

**关键词:** TCP SACK; 快重传和快恢复; 吞吐量模型; 时延带宽积

**中图分类号:** TP393

**文献标识码:** A

## Throughput model of TCP SACK under burst losses

WANG Zhi-ming, ZENG Xiao-ping, LI Juan, LIU Xue, CHEN Li

(College of Communication Engineering, Chongqing University, Chongqing 400044, China)

**Abstract:** Describing the end-to-end burst loss characteristics by Gilbert loss model, the model of RFC6675-based fast retransmission was proposed and recovery and a throughput model of TCP SACK based on the fast retransmission and recovery model was derived. The numerical and network simulations show that the fast retransmission and recovery model can accurately describe the fast retransmission and recovery process, the estimation accuracy of throughput model was promoted.

**Key words:** TCP SACK, fast retransmission and recovery, throughput model, bandwidth-delay product

### 1 引言

2012 年, IETF(Internet engineering task force) 在 TCP 协议中引入了基于 SACK(selective acknowledgment)的新分组丢失恢复机制——RFC6675, 以提升突发分组丢失下 TCP 的性能<sup>[1,2]</sup>。基于 SACK 的分组丢失恢复机制, 与 SACK 选项一起构成了基于 SACK 的快重传和快恢复机制, TCP 的 SACK 扩展已得到广泛部署和使用。TCP SACK 协议就是使用了 SACK 扩展的 TCP 协议。在采用 SACK 扩展的网络中, 目前还没有基于 RFC6675 的 TCP 吞吐量模型, 为准确评估采用 SACK 扩展的 TCP 协议在突发分组丢失下的性能, 建立突发分组丢失下基于 RFC6675 的 TCP SACK 吞吐量模

型非常必要<sup>[3,4]</sup>。

在现有的 TCP SACK 的吞吐量模型中, *pipe* 为快重传和恢复阶段网络中未被确认分组的估计数, 对快重传和快恢复阶段的分析大多都是基于在快恢复阶段每收到一个重复 ACK, *pipe* 变量就减小 1, 每收到一个部分 ACK, *pipe* 变量就减小 2 的假设<sup>[3,4]</sup>。然而, RFC 6675 中采用了 *pipe* 更新和新的分组丢失判定策略<sup>[1]</sup>: 1) *pipe* 中去除了已经被发端断定丢失的分组; 2) 一个分组是否被判为丢失取决于收到大于其序号分组的个数或总数据块的大小。这就造成分组的重传和恢复过程发生变化, 导致现有模型对基于 SACK 的快重传和快恢复过程描述不准确。另外, 现有模型对快重传和快恢复阶段的分析都忽略了在重传最后一个

收稿日期: 2015-02-05; 修回日期: 2015-06-14

基金项目: 国家自然科学基金资助项目 (No.61171089, No.61302054); 国家自然科学基金重大研究计划培育基金资助项目 (No.91438104)

**Foundation Items:** The National Natural Science Foundation of China (No.61171089, No.61302054), The Training Program of the Major Research Plan of the National Natural Science Foundation of China (No.91438104)

丢失分组前发送的新分组。实际上,在基于 SACK 的快重传和快恢复阶段,发端在重传第一个丢失和重传最后一个丢失之间可能发送新分组。在低随机分组丢失情况下,一个窗口发生多个分组丢失的概率非常小,这种发送新分组的情况可以忽略,但在突发分组丢失下,一个窗口中常常发生多个分组丢失,忽略发送新分组的情况就会造成对快重传和快恢复阶段发送分组数及其持续时间的计算不准确,从而影响 TCP SACK 吞吐量模型的准确性。

由于现有 TCP 吞吐量模型中已有大量关于标准 TCP 的建模和分析,因此,本文重点推导了快重传和快恢复阶段发送分组数及其持续时间的期望,即基于 RFC6675 的快重传和快恢复模型,并基于该模型建立了 TCP SACK 吞吐量模型。

## 2 突发分组丢失模型

近年来,网络分组丢失统计特性和相关研究一致表明<sup>[5-10]</sup>,互联网分组丢失率很小,但分组丢失具有突发性,且突发分组丢失率远高于网络平均分组丢失率,实时监测结果显示一些链路的突发分组丢失率甚至高于 10%<sup>[11]</sup>。Gilbert 模型是 Gilbert 提出的两状态马尔可夫链的差错模型,研究发现 89% 的网络分组丢失符合 Gilbert 模型<sup>[5]</sup>。近年来,基于 Gilbert 模型的分组丢失模型被广泛用于 TCP 吞吐量建模<sup>[5,10]</sup>。因此,本文采用 Gilbert 分组丢失模型描述 TCP 流的端对端突发分组丢失特性,并假设突发分组丢失率大于 5%。

Gilbert 分组丢失模型,如图 1 所示,有 2 个状态( $G$  和  $B$ )和 3 个参数( $p$ 、 $q$  和  $l$ )。 $B$  表示突发状态,在突发状态下分组丢失率为  $q$ , $G$  表示非突发状态, $G$  状态下分组不丢失,一旦发生分组丢失就会转入突发状态  $B$ ,其转移概率为  $p$ ,也称为突发事件率。 $B$  到  $G$  的转移概率为  $l$ ,由此可以得到突发持续时间, $E(B)=\frac{1}{l}$ 。

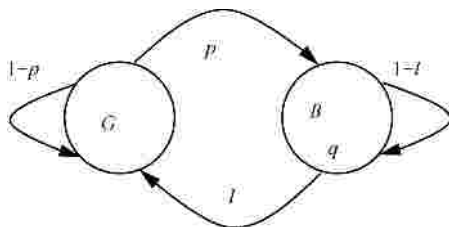


图 1 Gilbert 分组丢失模型

## 3 基于 RFC6675 的快重传和快恢复模型

基于 RFC6675 的快重传和快恢复建模就是计算快重传和快恢复阶段发送分组数的期望及其持续时间的期望。在基于 RFC6675 的快重传和快恢复机制中,接收端最多使用 4 个数据块告知发端已经收到不连续的数据块。发端根据每次收到的含有 SACK 信息的 ACK,判断窗口中分组的状态,在窗口允许条件下决定重传分组或发送新分组<sup>[1]</sup>。当发端检测到分组丢失并重传后,发端进入快恢复阶段,此时网络中最多还有  $W-?$  个分组。为实现快重传和快恢复建模,本节首先对快重传和快恢复阶段发送分组数的期望进行分析建模,然后根据发送分组数和持续时间的关系,得出持续时间的期望。本节用到的符号定义如表 1 所示。

符号	定义
$W$	发送分组丢失时的拥塞窗口大小
$q$	突发分组丢失率
$?$	分组丢失重传门限
$m$	窗口 $W$ 中的分组丢失数
$E_{FR}$	快重传和快恢复阶段发送分组数的期望
$D_{FR}$	快重传和快恢复阶段持续时间的期望
$R$	平均往返时延

与文献[4,5,10]中的假设类似,为方便推导,提出如下假设。

- 1) 重传分组不丢失。
- 2) 分组丢失数  $m < \frac{W}{2} - ?$ ,即分析中不考虑因分组丢失过多导致超时,因为这种超时不会进入快重传和快恢复。
- 3) 拥塞窗口  $W$  为最大突发长度。窗口  $W$  中除第一个丢失外,其他分组的丢失概率为突发分组丢失率  $q$ 。
- 4) 若序号为  $seq$  的分组未被确认而序号为  $seq+3$  的分组被选择性确认,则认为序号为  $seq$  的分组丢失。
- 5) 发端进入快重传前收到重复 ACK 不发送新分组。

### 3.1 快重传和快恢复模型及其近似简化

在重传分组不丢失和窗口中分组丢失数  $m < \frac{W}{2} - ?$  的情况下,不会发生超时。依据计数建模

方法<sup>[12]</sup>, 可将快重传和快恢复阶段发送分组数的期望  $E_{FR}$  的计算过程分为 3 部分: 快重传和快恢复阶段重传分组数的期望  $NR$ 、重传第一个丢失分组到重传最后一个丢失分组之间发送新分组数的期望  $N_1$  和重传最后一个分组到收到其确认之间发送新分组数的期望  $N_2$ , 即

$$E_{FR} = NR + N_1 + N_2 \quad (1)$$

1) 重传分组数的期望

在无超时的情况下, 发现第一个丢失时, 仍有  $W-2$  个分组在网络中, 其余  $m-1$  丢失发生在剩余的  $W-2$  个分组中。因此, 快重传和快恢复阶段重传分组数的期望等于窗口  $W$  中的丢失分组数的期望, 即  $NR = E[m]$

$$= \sum_{m=2}^{W-g} (m-1) C_{W-1}^{m-1} q^{m-1} (1-q)^{W-m} + 1 \approx 1 + (W-1)q \quad (2)$$

2)  $N_2$  的计算

当发端重传最后一个丢失分组后, 每次收到一个重复 ACK 或部分 ACK,  $pipe$  减小 1, 这就允许发端发送一个新分组(拥塞窗口始终为  $\frac{W}{2}$ )。直到收到最后一个重传分组的 ACK, 发端可以发送的新分组数为  $\frac{W}{2} - 1$ , 这个阶段发送分组数与丢失数  $m$  无关。因此, 重传最后一个分组到收到其确认之间发送新分组数的期望为

$$N_2 = \frac{W}{2} - 1 \quad (3)$$

3)  $N_1$  的推导

为计算  $N_1$ , 先考虑当  $m$  个丢失发生在前  $\frac{W}{2} + m - 2 + 1$  个分组中, 则在发现分组丢失时, 即窗口  $W$  中第一个分组丢失, 发端在重传最后一个丢失前最多收到的  $\frac{W}{2}$  个重复的 ACK, 最多使  $pipe$  减小到  $\frac{W}{2}$ 。因此, 发端在重传最后一个丢失分组前不会发送新分组。这种情况的概率为

$$C_{\frac{W}{2}+m-g}^{m-1} q^{m-1} (1-q)^{W-m} \quad (4)$$

当  $m$  个丢失在前  $\frac{W}{2} + m - 2 + i$  ( $2 \leq i \leq \frac{W}{2} - m + 2$ ) 个分组内但不全在前  $\frac{W}{2} + m - 2 + i - 1$  个分组内时, 即  $W$

个分组中的第  $\frac{W}{2} + m - 2 + i$  个分组丢失, 同时其余  $m-2$  个丢失发生在余下的  $\frac{W}{2} + m - 2 + i - 2$  个分组中, 在重传最后一个丢失前, 发端可以发送  $i-1$  个新分组。这种情况的概率为

$$C_{\frac{W}{2}+m-g+i-2}^{m-2} q^{m-1} (1-q)^{W-m}, \quad 2 \leq i \leq \frac{W}{2} - m + g \quad (5)$$

当  $m > \frac{W}{2}$  时, 第 1 个重传周期内发端没有机会发送新分组。从收到第 1 个重传分组的 ACK 开始(第 2 重传周期), 发端在重传最后一个丢失前不会再发生新分组。当  $m=1$  时, 发端只重传一个分组。因此, 当  $m > \frac{W}{2}$  和  $m=1$  时, 重传第一个丢失和最后一个丢失之间发送新分组数为 0。

综上所述, 在重传第一个分组丢失和重传最后一个分组丢失间发端发送的新分组数的期望  $N_1$  为

$$N_1 = \sum_{m=2}^{\frac{W}{2}} \left( \sum_{i=2}^{\frac{W}{2}-m+g} (i-1) C_{\frac{W}{2}+m-g+i-2}^{m-2} q^{m-1} (1-q)^{W-m} \right) \quad (6)$$

将式(2)、式(3)、式(6)代入式(1)即可得到基于 SACK 的快重传和快恢复阶段发送新分组数的期望。

$$E_{FR} = (W-1)q + \frac{W}{2} + \sum_{m=2}^{\frac{W}{2}} \left( \sum_{i=2}^{\frac{W}{2}-m+g} (i-1) C_{\frac{W}{2}+m-g+i-2}^{m-2} q^{m-1} (1-q)^{W-m} \right) \quad (7)$$

在基于 SACK 的快重传和快恢复中, 由于  $pipe$  始终被拥塞窗口  $\left(\frac{W}{2}\right)$  限制, 发端每一重传周期只能发送  $\frac{W}{2}$  个分组。另一方面, 在快重传和快恢复阶段, 就算只有一个丢失, 快重传和快恢复至少持续一个重传周期。因此, 可以由  $E_{FR}$  得到  $D_{FR}$  为

$$D_{FR} = \max \left( \frac{E_{FR}}{E[W]}, R, R \right) \quad (8)$$

$E_{FR}$  表达式中的  $N_1$  过于复杂, 为易于将  $E_{FR}$  用于 TCP 吞吐量建模, 需对  $N_1$  进行简化( $g=3$ )。图 2

通过三维曲面给出了  $N_1$  随窗口  $W$  和突发分组丢失率  $q$  的变化情况。在  $W$  不变情况下,  $N_1$  随  $q$  先增大后减小 ( $q > 0.5$  时  $N_1$  近似为 0)。当  $0.05 < q < 0.5$  时, 在  $q$  不变的情况下,  $N_1$  随窗口  $W$  逐渐增大。图 3 为当  $0 < q < 0.5$  时,  $N_1$  随  $q$  的变化情况。当  $0.05 < q < 0.5$  时,  $N_1$  随  $q$  线性减小。图 4 为当  $0.05 < q < 0.5$  时,  $N_1$  随  $W$  的变化情况。当  $0.05 < q < 0.5$  时,  $N_1$  随  $W$  线性增大。因此, 当  $0.05 < q < 0.5$  时,  $N_1$  可利用  $q$  和  $W$  的线性函数逼近。在  $q > 0.5$  情况下, 由式(6)易得  $N_1$  近似为 0。在突发分组丢失率  $q > 5\%$  的假设下, 采用回归分析法得到  $N_1$ , 如式(9)所示, 简化结果与如图 5 所示, 其简化误差小于 1%。

$$N_1 \approx \begin{cases} -2.41 + 0.99(0.5 - q)W, & 0.05 < q < 0.5 \\ 0, & 0.5 < q < 1 \end{cases} \quad (9)$$

将式(9)代入式(1), 且  $g=3$ , 得

$$E_{FR} = NR + N_1 + N_2 \approx \begin{cases} 0.995W + (0.01W - 1)q - 2.41, & 0.05 < q < 0.5 \\ (W - 1)q + 0.5W, & 0.5 < q < 1 \end{cases} \quad (10)$$

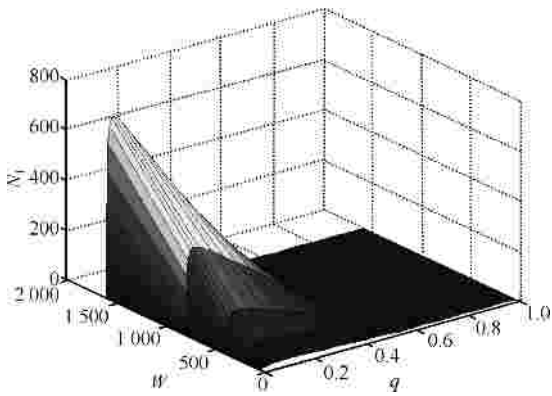


图 2  $N_1$  与  $W$  和  $q$  的关系

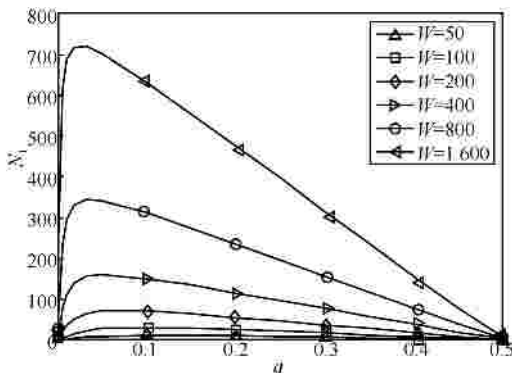


图 3  $N_1$  随分组丢失率  $q$  的变化

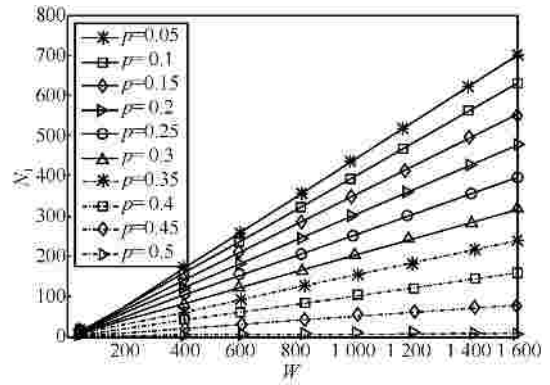


图 4  $N_1$  随窗口  $W$  的变化情况( $q > 0.01$ )

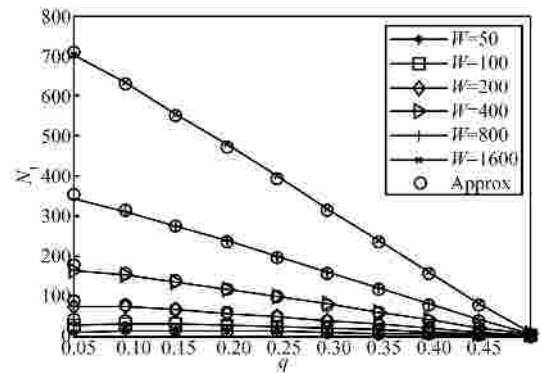


图 5  $N_1$  简化结果比较

$$D_{FR} = \frac{2E_{FR}}{W} R = \begin{cases} \frac{1.99W + (0.02W - 2)q - 4.82}{W} R, & 0.05 < q < 0.5 \\ \frac{2q(W - 1) + W}{W} R, & 0.5 < q < 1 \end{cases} \quad (11)$$

至此, 基于 RFC6675 的快重传和快恢复阶段发送分组数及其持续时间的期望就已得出, 即得到了基于 RFC6675 的快重传和快恢复模型(SFRR)。

### 3.2 SFRR 模型数值实验

采用蒙特卡罗法通过软件模拟基于 RFC6675 的快重传和快恢复过程, 并统计快重传和快恢复过程中发送分组的平均个数, 并与模型估计结果对比。TST 为采用部分 ACK 策略的吞吐量模型<sup>[3,4]</sup>对  $E_{FR}$  和  $D_{FR}$  的估计, Sim 为给定突发分组丢失率下经过 5 000 次实验的平均值。

图 6 所示为在基于 RFC6675 的快重传和快恢复阶段发送分组数的期望  $E_{FR}$  随突发分组丢失率  $q$  的变化。SFRR 模型结果与模拟实验统计值相符合, 而 TST 模型只有在  $q=0$  和  $q > 0.5$  时才与基于 RFC6675 的快重传和快恢复发送分组数的期望相符。

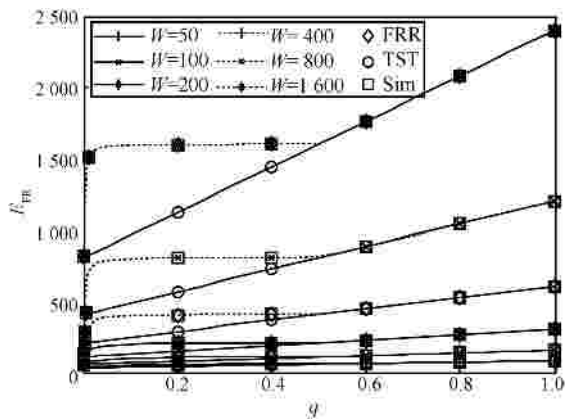


图 6  $E_{FR}$  随突发分组丢失率  $q$  的变化

图 7 所示为基于 RFC6675 的快重传和快恢复的平均持续时间  $D_{FR}$  随突发分组丢失率  $q$  的变化情况。由图可见，当  $q$  为 0~0.5 时，SFRR 预测的  $D_{FR}$  从 1 个往返时延( $R$ )迅速增加到逐渐逼近 2 个  $R$ ；当  $q$  为 0.5~1， $D_{FR}$  逐渐增加到 3 $R$ 。根据 RFC6675，在第 2 个重传周期开始时，所有分组丢失都会被发现。当分组丢失数小于  $\frac{W}{2}$ ，在 2 个  $R$  内就能重传完；当分组丢失数大于  $\frac{W}{2}$ ，则最多需要 3 个  $R$  恢复所有分组丢失。而 TST 等基于部分 ACK 策略的重传协议在第一个周期能够重传的分组数随着丢失数的增加而减小。当分组丢失增加时，其恢复时间快速增加；当分组丢失数多余  $\frac{W}{2} + 1$  (近似为  $q > 0.5$ )，则会由于重传完第一个分组丢失后不能发送新分组造成超时。

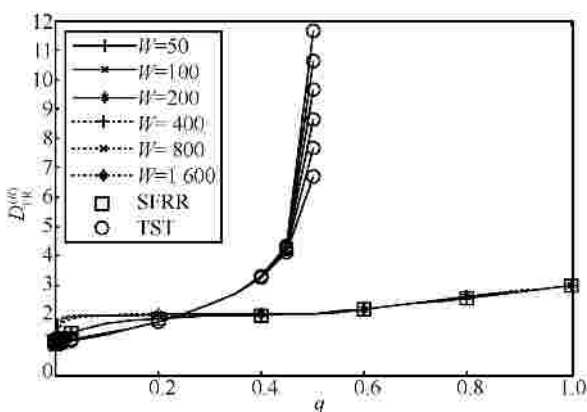


图 7  $D_{FR}$  随  $q$  的变化

由图 6 和图 7 可知，本文提出的模型能够准确估计基于 SACK 的快重传和快恢复阶段发送分组的平均数  $E_{FR}$  和平均持续时间  $D_{FR}$ ，现有吞吐量模型

不适用于对  $E_{FR}$  和  $D_{FR}$  的估计。

### 4 TCP SACK 吞吐量模型

不考虑超时 TCP 流的拥塞窗口的变化过程可以看作多个拥塞周期 (TDP) 的串联，一个 TDP 由一个拥塞避免阶段、一个快重传和一个恢复阶段组成的，TCP 吞吐量建模就是对这样一连串 TDP 建模的过程。TCP 吞吐量建模是对 TCP 流的稳态分析<sup>[4,5]</sup>，假设：1) TCP 流时间传输；2) 往返时延与拥塞窗口大小不相关；3) TCP SACK 的快重传和快恢复采用 RFC6675 中的分组丢失恢复机制；4) 在 Gilbert 分组丢失模型中，突发分组丢失率  $q > 5\%$ 。

将 TCP SACK 的吞吐量建模分为 2 步：1) 无超时情况下 ( $m < W - 3$ ) 的吞吐量模型 (NTO)；2) 考虑超时的完整吞吐量模型 FTM。记在一个 TDP 阶段内传输分组数的期望为  $S_{TDP}$ ，持续时间期望为  $D_{TDP}$ ，则其平均吞吐量为

$$T_{TDP} = \frac{S_{TDP}}{D_{TDP}} \tag{12}$$

#### 4.1 无超时模型 NTO

在不发生超时的情况下，发端至少收到 3 个重复 ACK。在已知突发事件率  $p$ ，突发分组丢失率  $q$  和突发长度  $E(B)$  情况下，则两次突发事件之间的发送分组数的期望  $E[a]$  为

$$E[a] = \sum_{k=1}^{\infty} (1-p)^{k-1} p^k = \frac{1}{p} \tag{13}$$

用  $E[d]$ 、 $E[m]$  和  $E[W]$  分别表示在一次突发过程中发送分组数的期望、丢失分组数的期望和突发长度的期望 (即最大拥塞窗口的均值)。当发现第一个分组丢失时，仍有  $E[W] - 1$  个分组在网络中。因此，余下的  $E[W] - 1$  个分组中包含  $E[m] - 1$  个丢失。假设  $E[m] - 1$  丢失服从均匀分布，则任意一次突发中分组丢失数的期望和在第一个丢失和最后一个丢失之间发送分组数的期望分别为

$$E[m] = 1 + (E[W] - 1)q \tag{14}$$

$$E[d] = E[W] - \frac{E[W] - 1}{2(E[m] - 1)} \tag{15}$$

由此可得到一个 TDP 周期中发送分组数的期望为

$$S_{TDP} = E[a] + E[d] \\ = \frac{1}{p} + E[W] - \frac{E[W]-1}{2(E[W]-1)} = \frac{1}{p} + E[W] - \frac{1}{2q} \quad (16)$$

$S_{TDP}$  和  $D_{TDP}$  也可以表示为

$$S_{TDP} = S_{LI} + S_b + S_{FR} - E[m] \quad (17)$$

$$D_{TDP} = D_{LI} + D_b + D_{FR} \quad (18)$$

其中,  $S_{LI}$  和  $D_{LI}$  表示发端在拥塞避免阶段发送分组数的期望和持续时间的期望;  $S_b$  和  $D_b$  分别表示从第一个丢失后的下一周期开始到快重传第一个丢失期间发送分组数的期望和持续时间的期望。由文献[4, 10]可知

$$S_{LI} = \frac{3}{8}E[W]^2 + \frac{3}{4}E[W] \quad (19)$$

$$D_{LI} = \left( \frac{E[W]}{2} + 1 \right) R \quad (20)$$

$$S_b = \frac{E[W]-1}{2} \quad (21)$$

$$D_b = \frac{R}{E[W]} S_b = \frac{R(E[W]-1)}{2E[W]} \quad (22)$$

令式(17)和式(18)的右边相等可以得到

$$\frac{1}{p} + E[W] - \frac{1}{2q} = S_{LI} + S_b + S_{FR} - E[m] \quad (23)$$

将式(10)对  $W$  求期望后即为  $S_{FR}$ , 与式(14)、式(19)和式(21)一起代入式(23), 忽略高阶项可求  $E[W]$  如下

$$E[W] \approx \begin{cases} \sqrt{\frac{24q-12p+84pq-40pq^2}{9pq}} + \frac{4q-5}{3}, & 0.05 < q < 0.5 \\ \sqrt{\frac{8q-4p+13pq}{pq}} - 1, & 0.5 < q < 1 \end{cases} \quad (24)$$

最后将式(15)和式(17)代入式(12), 并代入中间项可得到吞吐量的表达式如下

$$T_{NTO} \approx \frac{(2q-p)E[W] + 2E[W]^2 pq}{(E[W]^2 + 7E[W] + 0.04E[W]q - 4q - 10.64) pqR} \quad (25)$$

#### 4.2 完整吞吐量模型(FTM)

考虑超时的情况下, TCP 流传输过程可以看作一个个超时周期的串联, 每个超时周期包含了一个慢启动、多个 TDP 和一个超时阶段<sup>[4,10]</sup>。记  $p_{TO}$  为

超时概率,  $S_{SS}$  为慢启动阶段传送分组数的期望,  $D_{SS}$  为慢启动阶段持续时间的期望,  $D_{TO}$  为超时阶段持续时间的期望。则包含超时的 TCP SACK 吞吐量可表示为<sup>[4,12]</sup>

$$T_{Full} = \frac{(1-p_{TO})S_{TDP} + p_{TO}(S_{LI} + S_{SS})}{(1-p_{TO})D_{TDP} + p_{TO}(D_{LI} + D_b + D_{TO} + D_{SS})} \quad (26)$$

TCP SACK 在 2 种情况会导致超时: 1) 拥塞窗口  $W$  里丢失超过  $W-3$ ; 2) 快重传和快恢复阶段任意一个重传分组丢失。从快重传和恢复阶段开始, 分组丢失发生的概率为下一次突发事件发生的概率。因此

$$p_{TO} = \sum_{m=E[W]-2}^{E[W]} C(E[W]-1, m-1) q^{m-1} (1-q)^{E[W]-m} + \sum_{m=2}^{E[W]-3} (1-(1-p)^m) \quad (27)$$

从文献[4, 12]可以得到  $S_{SS}$ 、 $D_{SS}$  和  $D_{TO}$ , 将其代入式(26), 得到包含超时情况的 TCP SACK 吞吐量模型(FTM)

$$T_{Full} \approx \frac{\frac{1}{p} + E[W] - \frac{1}{2q} + \frac{p_{TO}}{1-p_{TO}} \left( 2^{\left(1+\frac{E[W]}{4}\right)} + \frac{3}{8}E[W]^2 + \frac{5}{4}E[W] \right)}{D_{TDP} + \frac{p_{TO}}{1-p_{TO}} \left[ \left( \frac{E[W]}{4} + \frac{E[W]+5}{2} \right) R + \frac{1+p}{1-p} RTO \right]} \quad (28)$$

$$\text{其中, } D_{TDP} = \frac{E[W]^2 + 7E[W] + 0.04E[W]q - 10.64 - 4q}{2E[W]} R。$$

## 5 仿真验证

### 5.1 仿真场景

在 NS2 下采用如图 8 所示的哑铃拓扑仿真场景,  $R_1$  和  $R_2$  之间的链路为瓶颈链路, 端对端往返时延约为 70 ms, 瓶颈带宽为 50 Mbit/s,  $C_1$  与  $S_1$  之间的时延带宽积(BDP)约为 7 Mbit/s, 属于大 BDP 网络场景。仿真时间为 200 s, 统计结果取自 50~200 s 的多次仿真结果的均值。Padhye<sup>[13]</sup>和 TST<sup>[3]</sup>模型的分组丢失率取平均分组丢失率。FTM 表示包含超时情况的 TCP SACK 吞吐量模型, 结果与 Parvez<sup>[10]</sup>等对比。

### 5.2 模拟分组丢失下仿真分析

分别采用 Gilbert 分组丢失模型和随机分组丢失模型在  $R_2$  到  $R_1$  的端口上产生随机分组丢失, 统计  $C_1$  与  $S_1$  之间 TCP SACK 流的吞吐量。图 9~图 11 分别为各模型在不同突发分组丢失率场景、不同

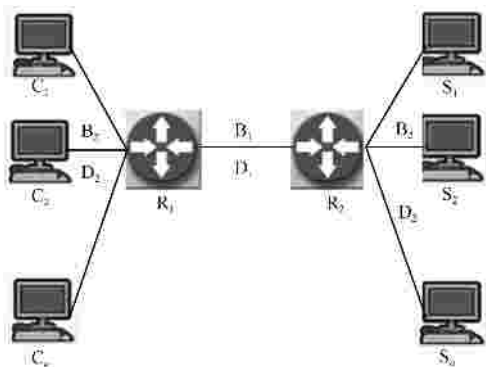


图 8 哑铃拓扑结构

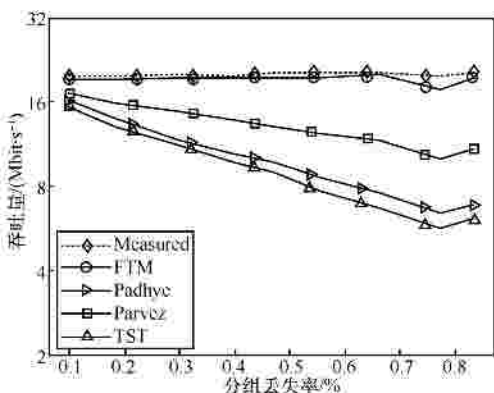


图 9 突发分组丢失率场景

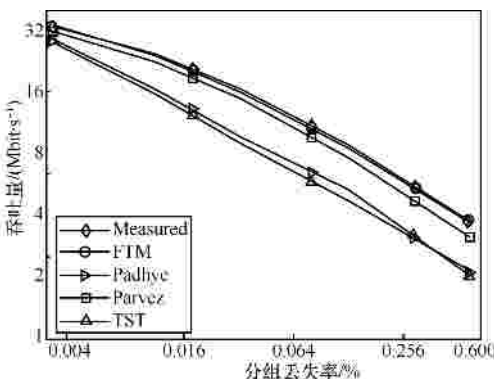


图 10 突发事件率场景

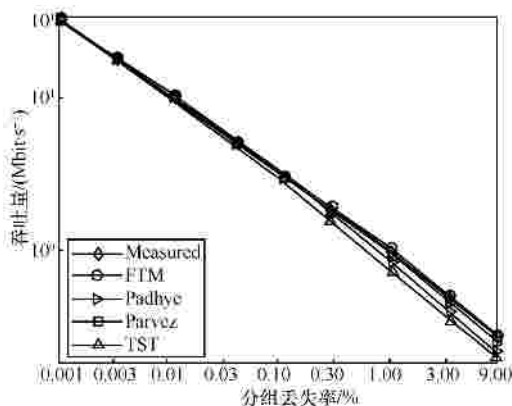


图 11 随机分组丢失率场景

突发事件率场景和随机分组丢失率场景仿真结果，图例 Measured 为 TCP SACK 流的吞吐量测量结果。

结果显示在 3 种情况下，FTM 能够准确地估计 TCP SACK 流的吞吐量。Parvez 在突发事件场景下估计误差较小，在突发分组丢失场景下，其估计误差逐渐增大。由于 Parvez 和 FTM 的计算都采用突发分组丢失率，其估计误差主要来自与 NewReno 的分组丢失恢复算法和 TCP SACK 的差异，随着突发分组丢失率增加，其性能差别越明显。Padhye 和 TST 在突发场景下均不能准确估计 TCP SACK 流的吞吐量。原因是在突发分组丢失模型下，总分组丢失率大于突发事件率，而 Padhye 和 TST 的计算都基于总分组丢失率，导致其吞吐量估计值远小于测量值。图 9 和图 10 的对比也表明，突发事件率对 TCP SACK 流吞吐量的影响要大于突发分组丢失率。在随机分组丢失场景下(如图 11 所示)，当分组丢失率较小时，几种模型的估计值都比较接近测量值。当随机分组丢失率较大时(>1%)，Parvez 和 FTM 的估计值好于 Padhye 和 TST。因为当随机分组丢失率较小时(多为单个分组丢失)，几种协议的分组丢失恢复过程一样。当分组丢失率很大时，与突发分组丢失的情况类似。图 9~图 11 的结果表明，FTM 能够准确估计随机分组丢失和突发分组丢失下的 TCP SACK 流吞吐量，Padhye 和 TST 不适用于高分组丢失率或突发分组丢失下的吞吐量估计，Parvez 不能于准确估计突发丢率下 TCP SACK 的吞吐量。

### 5.3 HTTP/FTP 和 UDP 背景流下仿真分析

实验场景采用 40 个 HTTP 流/10 个 FTP 流和 10 个 UDP 流分别作为背景流，HTTP 与 FTP 业务占比为 4:1，HTTP 业务请求间隔服从均值为 0.5 的负指数分布，网页大小服从均值为 48 KB 的 Pareto 分布，UDP 采用 1 Mbit/s 的 CBR 业务，背景业务流随机采用 TCP NewReno 或 TCP SACK 协议传输。

图 12 是在大 BDP 网络中当 HTTP/FTP 流的数量为 50~90 时 TCP SACK 流的吞吐量和模型估计值。当 HTTP/FTP 流数增加时，TCP SACK 流的吞吐量逐渐减小，突发分组丢失率逐渐增大。这是由于在瓶颈带宽不变的情况下，当 TCP 流个数增加时，TCP 流的平均带宽减小了。图 12 也表明模拟场景下网络分组丢失具有突发特性，突发分组丢失率在 0.228~0.455。当 HTTP/FTP 流数增加时，FTM 都能够准确估计 TCP SACK 流的吞吐量，Parvez 的

估计值的估计误差逐渐增加，Padhye 和 TST 的估计值偏离测量值较远。这与在模拟分组丢失模型下的结果一致，Parvez 的估计误差主要是由于在突发分组丢失下，TCP SACK 分组丢失恢复算法与 NewReno 的不同；Padhye 和 TST 不适用于突发分组丢失下 TCP SACK 的吞吐量估计。

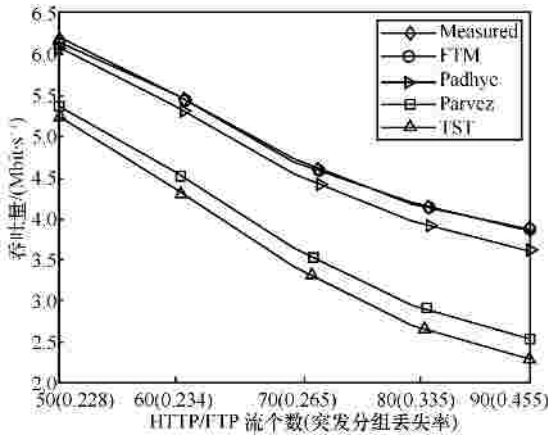


图 12 HTTP/FTP 背景流场景

图 13 是当 UDP 流的数量为 10~40 时 TCP SACK 流的吞吐量和模型估计值。当 UDP 流数增加时，TCP SACK 流的吞吐量逐渐减小，突发分组丢失率减小。由于增加恒定比特率的 UDP 流导致瓶颈可用带宽减小，TCP 流的平均带宽减小，限制了业务的突发性。结果显示，FTM 都能够准确估计 TCP SACK 流的吞吐量，Parvez 的估计误差逐渐减小，Padhye 和 TST 的估计值仍然偏离较远。这也说明突发分组丢失率降低，基于突发分组丢失的吞吐量模型和非突发分组丢失的吞吐量模型趋于一致，且不同分组丢失恢复算法性能差别减小。

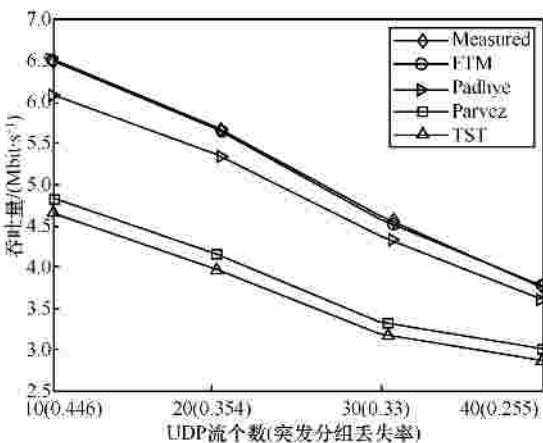


图 13 UDP 背景流场景

基于非突发分组丢失模型的 TCP 吞吐量模型中分组丢失是网络拥塞的标识，在基于突发分组丢失模型的 TCP 吞吐量模型中突发事件为拥塞标识。在突发分组丢失的网络中，网络总分组丢失率高于突发事件率，即网络拥塞的频率。Padhye 和 TST 在突发分组丢失和大 BDP 网络中，其估计的吞吐量低于实际的吞吐量就是由于这个原因。另一方面，随机分组丢失模型可以通过突发长度为 1 的突发分组丢失模型模拟。因此，基于突发分组丢失模型的吞吐量模型可用于非突发分组丢失的网络，但该模型是基于突发分组丢失率大于 5% 的假设，因此，当突发分组丢失率远低于 5% 时，可能不准确。

### 6 结束语

本文利用 Gilbert 分组丢失模型描述端对端突发分组丢失特性，推导了基于 RFC6675 的快重传和快恢复模型，并基于该模型建立了 TCP SACK 吞吐量模型。基于 RFC6675 的快重传和快恢复模型给出了基于 RFC6675 的快重传和快恢复阶段发送数据分组数及其持续时间的期望，该模型只与进入快重传时的拥塞窗口大小、突发分组丢失率和往返时延有关，但该模型没有考虑分组全部丢失或重传分组丢失造成超时的情况。其近似表达式是基于高突发分组丢失率的假设的，并不适用于只有随机分组丢失的情况。TCP SACK 吞吐量模型给出了 TCP SACK 吞吐量与突发分组丢失率、突发事件率、往返时延和超时时延的关系，与其他吞吐量模型类似，该模型是基于 TCP 流长时间传输的假设，即 TCP 流传输的稳态吞吐量模型，用于短时 TCP 吞吐量分析可能会存在误差。另外，根据网络分组丢失特性，模型中假设突发分组丢失率大于 5%，如果突发分组丢失率远低于 5%，结果可能会不准确。

数值实验验证了基于 RFC6675 的快重传和快恢复模型和其近似表达式的准确性。模拟分组丢失模型场景和不同背景流场景的实验表明，TCP SACK 吞吐量模型能够准确估计 TCP SACK 流的吞吐量。现有 TCP 吞吐量模型 (Padhye、Parvez 和 TST) 均不适用于估计突发分组丢失网络下 TCP SACK 流的吞吐量。

### 参考文献：

[1] BLANTON E, ALLMAN M, WANG L, et al. A conservative loss recovery algorithm based on selective acknowledgment (SACK) for

TCP[EB/OL]. <http://tools.ietf.org/html/rfc6675>, RFC 6675, 2012.

[2] FAN Y, AMER P. Non-renegable selective acknowledgments (NR-SACKs) for MPTCP[C]//International Conference on Advanced Information Networking and Applications Workshops. Barcelona, c2013: 1113-1118.

[3] 徐伟. TCP 协议的性能建模研究[D]. 安徽: 中国科学技术大学, 2012.

XU W. Study on performance modeling of TCP protocol[D]. Anhui, University of Science and Technology of China, 2012.

[4] SHEU T L, WU L W. An analytical model of fast retransmission and recovery in TCP-SACK[J]. Performance Evaluation. 2007, 64(6): 524-546.

[5] 曾彬, 张大方, 黎文伟, 等. 基于 Gilbert 分组丢失机制的 TCP 吞吐量模型[J]. 电子学报, 2009, (08): 1728-1732.

ZENG B, ZHANG D F, LI W W, et al. A TCP throughput model based on gilbert packet loss pattern[J]. Chinese Journal of Electronics, 2009, 37(8): 1728-1732.

[6] MEHMOOD M A, SARRAR N, UHLIG S, et al. Impact of access bandwidth on packet loss: a flow-level analysis[C]//IEEE Malaysia International Conference on Communications (MICC). Kuala Lumpur, c2013: 259-264.

[7] MEHMOOD M A, SARRAR N, UHLIG S, et al. Understanding flow performance in the wild[C]//Global Communications Conference (GLOBECOM). Atlanta, GA, c2013: 1410-1415.

[8] HASSLINGER G, SCHWAHN A, HARTLEB F, et al. 2-state(semi-) Markov processes beyond Gilbert-Elliott: traffic and channel models based on 2nd order statistics[C]//2013 Proceedings IEEE Infocom. Turin, c2013: 1438-1446.

[9] WANG W, KATO T, WU C, et al. Behavior analysis of different TCP implementations under burst error environment[R]. IEICE Tech Rep, Tokyo City University. c2014: 17-22.

[10] PARVEZ N, MAHANTI A, WILLIAMSON C. An analytic throughput model for TCP newReno[J]. IEEE/ACM Transactions on Networking, 2010, 18(2): 448-461.

[11] Internet health report[EB/OL]. <http://www.internetpulse.net/>, 2015.

[12] BERTRAN R, GONZÁLEZ M, MARTORELL X, et al. Counter-based power modeling methods: top-down vs bottom-up[J]. The Computer Journal, 2013, 56(2): 198-213.

[13] PADHYE J, FIROIU V, TOWSLEY D, et al. Modeling TCP Reno performance: a simple model and its empirical validation [J]. IEEE/ACM Transactions on Networking, 2000, 8(2): 133-145.

作者简介：



王志明 (1986-), 男, 陕西永寿人, 重庆大学博士生, 主要研究方向为 TCP 拥塞控制与建模分析、航空自组网组网及优化。



曾孝平 (1956-), 男, 四川广安人, 重庆大学教授、博士生导师, 主要研究方向为航空移动通信、计算机网络、信号与信息处理。



李娟 (1989-), 女, 云南大理人, 重庆大学硕士生, 主要研究方向为航空自组网 MAC 协议及资源调度算法。



刘学 (1983-), 男, 重庆人, 重庆大学博士生、讲师, 主要研究方向为航空自组网连通性研究、计算机网络。



陈礼 (1982-), 男, 重庆人, 重庆大学博士生、讲师, 主要研究方向为民用航空移动通信网络。